Correlated Topics in a Scalable Multidimensional Text Cube: Algorithms and Aviation Safety Case Study

Bo Zhao[1] Cindy X. Lin[1] Ashok N. Srivastava[2] Nikunj C. Oza[2] Jiawei Han[1]

[1]University of Illinois at Urbana-Champaign
{bozhao3, xidelin2}@uiuc.edu, hanj@cs.uiuc.edu
[2]NASA Ames Research Center
{ashok.n.srivastava, nikunj.c.oza}@nasa.gov

ABSTRACT

As world-wide air traffic continues to grow even at a modest pace, the overall complexity of the system will increase significantly. This increased complexity can lead to a larger number of fatalities per year even if the extremely low fatality rate that we currently enjoy is maintained. One important source of information about the safety of the aviation system is in Aviation Safety Text Reports which are written by members of the flight crew, air traffic controllers, and other parties involved with the aviation system. These anonymized narrative reports contain fixed-field contextual information about the flight but also contain free-form narratives that describe, in the author's own words, the nature of the safety incident and, in many cases, the contributing factors that led to the safety incident. Several thousand such reports are filed each month, each of which is read and analyzed by highly trained experts. However, it is possible that there are emerging safety issues due to the fact that they may be reported very infrequently and in different contexts with different descriptions. The goal of this research paper is to develop correlated topic models which uncover correlations in the subspaces defined by the intersection of numerous fixed fields and discovered correlated topics. This task requires the discovery of latent topics in the text reports and the creation of a topic cube. Furthermore, because the number of potential cells in the topic cube is very large, we discuss novel methods of pruning the search space in the topic cells, thereby making the analysis feasible. We demonstrate the new algorithms on an analysis of pilot fatigue and its contributing factors, as well as the safety incidents that are correlated with this phenomenon.

Keywords

Aviation Safety Database, Topic Correlation Analysis, Text Cube

1. INTRODUCTION

Many organizations have large text repositories that contain information that is mission critical to the organization. NASA, for example, operates a safety reporting system known as the Aviation Safety Reporting System (ASRS) which collects voluntarily submitted aviation safety incident/situation reports from pilots, controllers, and others with the purpose to identify system-wide deficiencies and safety issues[5]. ASRS can receive as many as several thousand reports in a month and contains over 100,000 reports at this time. The analysts at ASRS analyze each report in detail and assign reports to potentially several of 60 high-level anomaly categories and conduct other safety related studies with the reports. The reports also contain various 'fixed-field' pieces of information that identify the context and operating conditions of the flight. The reports are anonymous by law; thus, the author, his or her organization, and other identifying pieces of information are removed from the reports. The ASRS analysts also use automated tools to help them study the reports and compare them with others, but the vast majority of the work is done by skilled experts.

In the early 1990's, ASRS personnel issued an alert to the Federal Aviation Administration (FAA) based on studies that were made on text reports submitted over the previous several years. This study concluded that the Boeing 757 generated wake-turbulence, a form of turbulence that is created behind an aircraft as it passes through the air. This dangerous phenomenon can lead to catastrophic consequences for smaller aircraft that are following the lead aircraft and is an important factor in determining the capacity of an airport [3].

When the B757 was initially put into service it had a wake-vortex classification that allowed a smaller separation between it and a trailing aircraft. The alert issued in the late 1980's and 1990's was noted by the FAA but, as it turns out due to unrelated reasons, they took action on reclassifying the 757 aircraft into one that requires a larger aircraft separation after two fatal accidents. Thus, although the regulatory agency didn't take action on this particular alert, it is an excellent example of the identification of precursors to catastrophic accidents based on the analysis of text reports.

In principle, to detect and find corroborating evidence of this problem, the ASRS analysts had to comb through a huge amount of information in their system to detect and document this problem. The research discussed in this pa-

SIGKDD'10, July 25–28, 2010, Washington, DC, USA.

per addresses the problem of providing automated methods to automatically identify potential precursors to safety incidents in large volumes of safety related reports using a combination of a correlated topic model and a powerful and scalable multidimensional cube.

Several methods have been developed that can help enable the automatic classification [19] of text reports into anomaly categories [18] [15] and significant work has been performed in the area of correlated topic models [1].

The problem that we focus on in this paper is the generation of a method to automatically uncover documents that are correlated with a topic of interest and then analyze the resulting set of reports using a scalable multidimensional cube. The multidimensional cube could consist of the fixed-fields already identified in a set of reports, but more interestingly, other topics that have been discovered in the text repository. These reports can offer a significant amount of insight into the main topic and its contributing factors. We use this as a running example throughout the paper to illustrate the performance and output of the system.

For example, consider a study of pilot fatigue, which is thought to be a contributing factor to aviation safety incidents. The authors may not directly mention the word *fatigue* in their writeups. Instead, they may mention other phrases such as *I was on the last leg of a 5 segment trip"* or *THIS WAS THE FINAL LEG OF A MULTI-LEG FLT AND I WAS MORE TIRED THAN I THOUGHT* [20]. In these examples, the author does not directly state the word fatigue. In the last example, the author indicated that he/she was tired due to being on the final leg of a trip. Notice in this excerpt that the author uses abbreviations; ASRS documents are laden with abbreviations in the narrative sections.

2. PROBLEM FORMULATION

2.1 Preliminaries: The Text Cube Model

A set of documents D is stored in an n-dimensional database $DB = (A_1, A_2, \cdots, A_n, D)$. Each row $r \in DB$ corresponds to a document $d \in D$ in the form of $r = (a_1, a_2, \cdots, a_n, d)$, where $a_i \in A_i$ means the value of the dimension A_i for r is a_i. We denote $r(D) = d$ and $r(A_i) = a_i$.

The data cube model [8] extended to the above multidimensional text database is called the text cube [13]. Several important concepts are introduced as follows.

Definition 1. ***Text Cube: Cell and Measure***. *In the text cube built on a set of documents D, a* ***cell*** *is in the form of $c = (a_1, a_2, \cdots, a_n : D', f_1(D'), f_2(D'), \cdots, f_m(D'))$, where either $a_i \in A_i$ (i.e., the value of dimension A_i for c is a_i) or $a_i = *$ (i.e., the dimension A_i is aggregated in c). D' is the* aggregated document set *for c, formally defined as $D' = \{r \mid r \in DB, r(A_i) = a_i \; if \; a_i \neq *\}$. $f_1(D'), f_2(D'), \cdots, f_m(D')$ are* measures *on D' that are computed by* aggregate functions. *We denote $c(D) = D'$ and $c(A_i) = a_i$.*

Cells with m non-$*$ dimensions are called m-dim cells. An n-dim cell is said to be a base cell, with no aggregated dimension, and a 0-dim cell is the apex cell that aggregates all dimensions.

Definition 2. ***Ancestor and Descendant***. *Cell c' is an ancestor of c (or c is a descendant of c') iff $\forall \; i : c'(A_i) \neq * \Rightarrow c(A_i) = c'(A_i)$. Note a cell is an ancestor (or descendant) of itself. A base cell has no descendant except itself, and the apex cell has no ancestor except itself.*

Definition 3. ***Parents and Children*** *are immediate ancestors and descendants of a cell, respectively. Cell c' is a parent of c (or c is a child of c') iff (i) c' is an ancester of c, and (ii) c' is an i-dim cell while c is an $(i+1)$-dim cell.*

Measures in a text cube are categorized into *distributive*, *algebraic*, and *holistic* [10] based on the way of aggregate functions used.

Distributive: An aggregate function is distributive if the aggregate value of a cell c can be computed by only using the aggregate values of c's children, *e.g.*, count() and sum().

Algebraic: An aggregate function is algebraic if it can be computed by an algebraic function on a limited number of distributive measures, *e.g.*, avg() and deviation().

Holistic: An aggregate function is holistic if there is no constant bound on the storage size needed to describe a subaggregate, *e.g.*, median() and mode().

To make the time and space complexity affordable, in most cases, we require a measure for a text cube to be either distributive or algebraic.

Definition 4. ***Topic***. *A semantically coherent topic in a text collection is represented by a topic model θ, which is a probabilistic distribution of words $\{p(w|\theta)\}_{w \in W}$. W is the vocabulary. Clearly, we have $\sum_{w \in W} p(w|\theta) = 1$.*

2.2 Topic Correlation Analysis in Text Cube

2.2.1 Problem

We define the task of topic correlation analysis in text cube as follows. Given a text cube and k topics, we aim to answer the following two questions:

Topic Relevance Analysis. Given a keyword query $Q = \{q_1, q_2, \cdots, q_{|Q|}\}$, what is the most relevant topic θ to Q?

Topic Correlation Analysis. Given a cell c, are there other topics $\alpha \neq \theta$ such that θ and α are correlated in the scenario of c? The *scenario* of cell c is a condition, represented by a selected set of dimensions, with some possibly instantiated, such as Weather = "Fog", on which the topic correlations will be analyzed.

EXAMPLE 1. *$k = 3$ topics are generated and the word distributions of topics are shown in Table 1. Notice that*

Topic 1		Topic 2		Topic 3	
day	0.057	engine	0.230	factor	0.040
hour	0.043	oil	0.004	awareness	0.015
trip	0.027	shutdown	0.013	lack	0.011
time	0.027	pressure	0.012	fail	0.011
rest	0.019	start	0.011	performance	0.009
night	0.019	power	0.007	corrective	0.009
leg	0.017	temperature	0.007	attention	0.007
fatigue	0.012	landing	0.005	error	0.007
morning	0.009	compressor	0.005	action	0.007
long	0.009	viberation	0.004	realize	0.007
early	0.007	restart	0.003	poor	0.006
tired	0.007	fail	0.003	failure	0.006
sleep	0.007	filter	0.003	miss	0.005

Table 1: Word Distributions.

	Dimensions			Topic Distributions		
Doc	State	Time	Weather	Topic 1	Topic 2	Topic 3
d_1	IL	night	rain	0.3	0.1	0.6
d_2	IL	night	snow	0.6	0.2	0.2
d_3	CA	night	snow	0.2	0.4	0.4
d_4	CA	daytime	snow	0.4	0.3	0.3
d_5	NY	night	snow	0.1	0.8	0.1
d_6	NY	daytime	rain	0.5	0.4	0.1
d_7	NY	daytime	fog	0.3	0.3	0.4

Table 2: Text Cube and Topic Distributions.

the first topic contains many words that are related to being tired or fatigued. The second topic contains words that describe potential issues in an engine, and the third topic contains words that are related to attention and awareness. In Table 2, a text cube is built on the document set $D = \{d_1, d_2, \cdots, d_7\}$ *with three dimensions 'State', 'Time of the Day' and 'Weather' as well as the topic distributions of documents. Given the keyword query* $Q = \{$*'I','am','tired'*$\}$*, Topic 1 is regarded as the most relevant topic to* Q*. Given the cell* $c_1 = (*,$*'night'*$,*$)$*, Topic 3 is correlated to Topic 1 in the scenario of* c_1*. Also, given the cell* $c_2 = (*, *,$*'snow'*$)$*, Topic 2 is correlated to Topic 1 in* c_2*.*

2.2.2 Motivation

Topic relevance and correlation analysis in text cube is useful to aviation safety analysis for several reasons, including:

- Many aviation safety databases such as ASRS [1] consist of both textual (*e.g.*, the pilot report about the accident) and multi-dimensional (*e.g.*, 'location', 'time' and 'weather' associated with the pilot report) information, which can naturally fit in a text cube [13, 23].

- A topic in the aviation safety databases corresponds to an issue that that may explain what happened during the flight that caused the issue. For example, in Table 1, Topic 1 describes a 'fatigue' problem, which could be caused by 'long duration trip' or 'early awakening from sleep'. Topic 2 contains 'engine' issues, and Topic 3 describes the 'attention and awareness' related issues.

- Users of aviation safety databases may not have complete knowledge about a flight issue (*i.e.*, a topic), but instead use a set of keywords to express their target topic. Based on this type of input, Topic Relevance Analysis could supply a way to match user queries to underlying topics.

- To analyze correlated topics is to analyze correlated flight issues. The latter can facilitate or be a fundamental component for many aviation safety applications, such as classification [15], causal analysis [3] and error source finding [11].

[1] http://asrs.arc.nasa.gov/

Organization. We organize the rest of this paper as follows. Section 3 generates topics in the preprocessing step. Section 4 and Section 5 propose solutions to the Topic Relevance Analysis and the Topic Correlation Analysis problems respectively. Section 6 performs experimental studies on a real aviation safety database, and finally, Section 7 concludes the whole paper.

3. PREPROCESSING

Documents in aviation safety databases contain many abbreviations, acronyms and phrases. Abbreviations and antonyms can be transformed to their original complete formats by utilizing domain dictionaries, but phrases are much more difficult to handle because of the large number of potential combinations.

To overcome this problem, sequential pattern mining techniques are developed in [6], by which each detected pattern (*i.e.*, a set of keywords that appear together frequently) is regarded as a phrase and appearances of phrases in documents are replaced by special terms that stand for corresponding phrases.

After phrases are replaced by terms, k topics are generated by running LDA (Latent Dirichlet Allocation) [2]. For the rest of this paper, we assume both *word distributions over topics* (*i.e.*, $Pr(w|\theta)$ for a word w and a topic θ) and *topic distributions over documents* (*i.e.*, $Pr(\theta|d)$ for a topic θ and a document $d \in D$) are prior knowledge.

4. TOPIC RELEVANCE ANALYSIS

In this section, we formally define the problem of topic relevance analysis as: given a keyword query $Q = \{q_1, q_2, \cdots, q_{|Q|}\}$, which topic θ maximizes the relevance score $Rel(Q, \theta)$ based on Q:

$$Rel(Q, \theta) = Pr(\theta|Q) = \frac{Pr(Q|\theta)Pr(\theta)}{Pr(Q)} \quad (1)$$

4.1 Relevance Function

According to the theory of probability, we convert the second component in the numerator of Equation 1 to be as

$$Pr(\theta) = \sum_{d \in D} Pr(\theta|d)Pr(d),$$

where $Pr(d)$ is supposed to conform to uniform distribution, *i.e.*, $Pr(d) = \frac{1}{|D|}$.

Following unigram topic modeling algorithms [17, 12], we assume the independence among words, so that the first component in the numerator and the dominator of Equation 1 becomes:

$$\frac{Pr(Q|\theta)}{Pr(Q)} = \prod_{q_i \in Q} \frac{Pr(q_i|\theta)}{Pr(q_i)},$$

where $Pr(q_i)$ equals the occurence of q_i divided by the total occurence of all words in D, *i.e.*,

$$Pr(q_i) = \frac{count(q_i, D)}{\sum_{w \in W} count(w, D)}$$

Recall that W is the vocabulary for any topic.

Finally, Equation 1 turns out to be:

$$Rel(Q, \theta) = \left(\frac{1}{|D|} \sum_{d \in D} Pr(\theta|d) \right) \prod_{q_i \in Q} \frac{Pr(q_i|\theta)}{Pr(q_i)} \quad (2)$$

EXAMPLE 2. *Following Example 1, a keyword query is given as* Q = { '*tired*', '*long*', '*trip*'}, *and we calculate the relevance scores for each of the three topics. Since only words with the highest probabilities are listed, we simply assume the probability of an unlisted word in a topic equals to* 0.001. *Plus, for the prior of keywords, we have Pr('tired') = 0.01256, Pr('long') = 0.06675, and Pr('trip') = 0.05573. According to Equation 2, we get these relevance scores as Rel(Q, Topic 1) = 0.01248, Rel(Q, Topic 2) = 7.643e-6, and Rel(Q, Topic 3) = 6.688e-6, among which Topic 1 is the most relevance topic to the query* Q. *Note that the result is approximate because of the assumption that an unlisted word has a generative probability* 0.001.

4.2 Complexity Analysis

4.2.1 Complexity without Pre-computation

After the keyword query Q arrives, we scan the k topics one by one. For each topic θ, we use Equation 2 to calculate $Rel(Q, \theta)$ and output the best topic that maximizes the relevance score. The computational cost for the first and the second parts of Equation 2 are $O(|D|)$ and $O(|Q|)$, respectively. Hence, the overall computational complexity for exhausting all topics is

$$O(k(|D| + |Q|)).$$

An aviation safety database usually stores a vast amount of records, *e.g.*, in our ASRS dataset, we have 61,235 flight records for 10 years, hence the above time complexity is unsatisfactorily large for online queries.

4.2.2 Complexity with Pre-computation

It is easy to see that $Pr(\theta|d)$ is independent of queries, so we can pre-compute and store $Pr(\theta)$ for each topic θ, which results in reducing the overall computational cost to

$$O(k(1 + |Q|)) = O(k|Q|).$$

Note in this case, we need additional $O(k)$ space to store the pre-computation results. Usually, both k and $|Q|$ are small. For example, in our experimental study on the ASRS dataset, k is 100, and $|Q|$ is no more than 10 words. Hence, the efficiency is guaranteed to be good, which is generally fast enough to respond to any online query.

5. TOPIC CORRELATION ANALYSIS

In this section, the formula of topic correlation score is discussed in Section 5.1, and a dilemma regarding the computational issue is stated in Section 5.2. To overcome the dilemma, we propose the idea of partially materializing the text cube. Hence, Section 5.3 explains how to process queries in a partially materialized text cube, and Section 5.4 introduces how to select cells for pre-computation.

5.1 Correlation Score

In traditional topic correlation analysis [1, 16, 21], two topics are correlated if they have the same or similar context. The so-called 'context' is usually explained as a corpus. Formally, a typical way to define the correlation of two topics α and β over a corpus $D = \{d_1, d_2, \cdots, d_{|D|}\}$ is to calculate the angle of two vectors:

$$Col(\alpha, \beta) = Cosine(\overrightarrow{V}(\alpha), \overrightarrow{V}(\beta)) = \frac{\overrightarrow{V}(\alpha) \cdot \overrightarrow{V}(\beta)}{||\overrightarrow{V}(\alpha)|| \; ||\overrightarrow{V}(\beta)||},$$

where $\overrightarrow{V}(\alpha)$ is the topic distribution vector of α, *i.e.*,

$$\overrightarrow{V}(\alpha) = \left(Pr(\alpha|d_1), Pr(\alpha|d_2), \cdots, Pr(\alpha|d_{|D|})\right). \quad (3)$$

To consider the above topic correlation problem in the scenario of the text cube, we regard the 'context' to be cells. Concretely, for a cell c, we re-define Equation 3 as

$$\overrightarrow{V}_C(\alpha) = \left(Pr(\alpha|d'_1), Pr(\alpha|d'_2), \cdots, Pr(\alpha|d'_{|C(D)|})\right),$$

where $c(D) = \{d'_1, d'_2, \cdots, d'_{|c(D)|}\}$ is the aggregated document set for the cell c.

To sum up, the correlation score of two topics α and β in the cell c equals to

$$Col_C(\alpha, \beta) = \frac{\overrightarrow{V}_C(\alpha) \cdot \overrightarrow{V}_C(\beta)}{||\overrightarrow{V}_C(\alpha)|| \; ||\overrightarrow{V}_C(\beta)||} \quad (4)$$

EXAMPLE 3. *Following Example 2, consider three cells* c_1 = (∗, '*night*', ∗) *and* c_2 = (∗, ∗, '*snow*'). *For* c_1, *the topic distribution of the three topics* (0.3, 0.6, 0.2, 0.1), (0.1, 0.2, 0.4, 0.8) *and* (0.6, 0.2, 0.4, 0.1), *respectively. The correlation score between Topic 1 and Topic 2 is* 0.4755, *and the one between topic 1 and topic 3 is* 0.7305. *For* c_2, *the topic distribution of the three topics* (0.6, 0.2, 0.4, 0.1), (0.2, 0.4, 0.3, 0.8) *and* (0.2, 0.4, 0.3, 0.1), *respectively. The correlation score between Topic 1 and Topic 2 is* 0.5494, *and the score between Topic 1 and Topic 3 is* 0.7980. *It is observed that Topic 1 and Topic 3 are correlated in the scenarios of 'night' and 'snow'.*

5.2 Full Cube Computation

Since the number of topics (*i.e.*, k) is small, the problem of finding correlation topics could be split into first calculating $Col_C(\alpha, \theta)$ for each topic α ($\alpha \neq \theta$) and then outputting the best α that maximizes the correlation score. When the queried cell c comes, we can simply scan all documents in c and compute Equation 4 as:

$$Col_c(\alpha, \theta) = \frac{\sum_{d \in c(D)} Pr(\alpha|d) Pr(\theta|d)}{\sqrt{\sum_{d \in c(D)} Pr^2(\alpha|d)} \sqrt{\sum_{d \in c(D)} Pr^2(\theta|d)}}, \quad (5)$$

However, the computational cost is $O(|c(D)|)$, *i.e.*, number of documents in cell c, which is too large to guarantee in time response to online queries. To reduce the online computational cost, we can offline compute and store some values, so that answering online queries can be accelerated by utilizing stored values. This step is called *materialization* [8] of the data cube.

Concretely, we decompose Equation 5 into two parts:

1. $SS_\theta(c) = \sum_{d \in c(D)} Pr^2(\theta|d)$ for each topic, and
2. $DM_{\theta_1,\theta_2}(c) = \sum_{d \in c(D)} Pr(\theta_1|d) Pr(\theta_2|d)$ for each pair of topics.

For the convenience of expression, we abbreviate $\{SS_\theta(c)\}_\theta$ and $\{DM_{\theta_1,\theta_2}(c)\}_{\theta_1,\theta_2}$ as $SS(c)$ and $DM(c)$, respectively.

The simplest algorithm to compute measures in a full n-dimensional text cube is: first compute all cells in the n-D cuboid (*i.e.*, base cells); then compute all cells in the (n-1)-D cuboid; $\cdots$; finally compute the 0-D cuboid (*i.e.*, the apex cell). After the full text cube is computed, any topic correlation scores can be queried by directly retrieving $SS()$ and $DM()$ and doing some simple computation. However, the key points of such materialization are: (i) how much the storage cost is; and (ii) how an r-D cell is aggregated from some (r-1)-D cells without looking at the original database. We will discuss the two issues:

Storage cost. In tradition data cube, usually only $O(1)$ space is required by the measure of each cell; however, for $SS(c)$ and $DM(c)$, we need as much as $O(k^2)$ storage size. For aviation safety databases, problems (*i.e.*, topics) that happened during the flight are complex, diverse and variant. For example, in our experiments, there are as many as 100 topics in the ASRS datasets, whose range covers 'environmental facts', 'human factors', 'engine problem', *etc.*. Such special situations cause the sharp enlargement of the storage size compared to traditional data cubes. Although storage is more and more cheap, such space complexity is still excessive.

Aggregation. Both SS and DM are distributive measures [8]. Let $c = (a_1, a_2, \cdots, a_n : c(D), SS(c), DM(c))$ be an (r-1)-D cell. *W.l.o.g.*, suppose $a_n = *$ and A_n has m distinct values $a_{n,1}, a_{n,2}, \cdots, a_{n,m}$, so we have c's children as $c_j = (a_1, a_2, \cdots, a_{n,j} : c_j(D), SS(c_j), DM(c_j))$ for $j = 1, 2, \cdots, m$. It is easy to prove that $SS(c)$ and $DM(c)$ can be efficiently aggregated from $SS(c_j)$ and $DM(c_j)$ as

$$SS_\theta(c) = \sum_{c_j} SS_\theta(c_j)$$

$$DM_{\theta_1,\theta_2}(c) = \sum_{c_j} DM_{\theta_1,\theta_2}(c_j)$$

5.3 Query Processing in Partially Materialized Cube

Although SS and DM can be efficiently aggregated, unlike distributive/algebraic measures in traditional data cube, they consume a huge amount of space if materialized for all cells, which is not affordable for an aviation safety database. Therefore, in this subsection, we introduce how to process the topic correlation queries in a text cube where only a subset of cells are precomputed; and in Section 5.4, we discuss how to optimize the storage size by appropriately choosing the subset.

A text cube is said to be *partially materialized* if a subset of cells are precomputed while the rest are not. In such a text cube, a query should be processed as:

1. If the corresponding cell is precomputed, the value stored can be directly returned;
2. Otherwise, we can obtain this cell by aggregating a set of precomputed cells.

For a non-materialized cell, there are different ways of choosing the set of precomputed ones to obtain the inquired cell. So we have the chance to choose the 'optimal' set which incurs the minimum cost in the aggregation process. To formally define the query processing problem, we first need to introduce the concepts of *decision space* and *cost model*.

Decision Space. For a non-materialized queried cell $c = (a_1, a_2, \cdots, a_n : c(D), SS(c), DM(c))$. *W.l.o.g*, suppose $a_i = *$ for $i = 1, 2, \cdots, n'$, and $a_i \in A_i$ for $i = n'+1, n'+2, \cdots, n$. We furthermore denote $c_{i,j} = (a_1, a_2, \cdots, a_{i-1}, a_{i,j}, a_{i+1}, \cdots, a_n : c_{i,j}(D), SS(c_{i,j}), DM(c_{i,j}))$ for $i = 1, 2, \cdots, n'$ and $a_{i,j} \in A_i$. We have n' choices to aggregate c, *i.e.*, the so-called A_i-based aggregation is to aggregate c from the set of cells $\{c_{i,1}, c_{i,2}, \cdots, c_{i,|A_i|}\}$. *W.l.o.g.*, suppose we select to aggregate c based on A_1, then for each $c_{1,j}$, if it is pre-computed, it can be directly retrieved; otherwise, recursively, to obtain SS and DM for this cell, we have n'-1 choices to aggregating other cells on one of dimensions $A_2, A_3, \cdots, A_{n'}$.

Cost Model. Given a queried cell c, the cost of processing the query is the number of precomputed cells we need to access. Particularly, if c corresponds to an empty cell, the cost defined to be 0. If c corresponds to a precomputed cell, the cost is 1.

Now the query processing problem turns out to be: what is the best way to aggregate a queried cell c so that the cost is minimum? We use the dynamic programming algorithm $Aggregate(c)$ to recursively compute the optimal cost/decision. $Cost(c)$ denotes the optimal cost of a queried cell c, and $Best(c)$ denotes the corresponding set of cells that need to be accessed under the optimal cost. Of course, $|Best(c)| = Cost(c)$. We compute $Cost(c)$ and $Best(c)$ case by case:

1. $Cost(c) = 0$, if c corresponds to an empty cell. In this case, $Best(c) = \emptyset$, and we respond to the query by returning $SS_\theta(c) = 0$ for any topic θ and $DM_{\theta_1,\theta_2}(c) = 0$ for any pair of topics θ_1 and θ_2.
2. $Cost(c) = 1$, if c corresponds to a pre-computed cell. Here, $Best(c) = \{c\}$, and we answer the query by directly retrieving the stored values.
3. Let $i' = \arg\min_{i,c(A_i)=*} \left(\sum_{a_{i,j} \in A_i} Cost(c_{i,j}) \right)$, if c corresponds to an non-empty, non-materialized cell. In this situation,
$$Cost(c) = \sum_{a_{i',j} \in A_{i'}} Cost(c_{i',j})$$
$$Best(c) = \bigcup_{a_{i',j} \in A_{i'}} Best(c_{i',j})$$
If $c_{i',j}$ is not materialized, we repeat the same procedure.

Algorithm 1 shows the pseudo code of $Aggregate(c)$.

Algorithm 1 Aggregate(c)

```
ALGORITHM:
if c is empty then
  Cost(c) ← 0; Best(c) ← ∅;
  return ;
end if;
if c is materialized then
  Cost(c) ← 1; Best(c) ← {c};
end if;
Cost(c) ← +∞;
for each i s.t. c(A_i) = * do
  CurCost = 0;
  for each a_{i,j} ∈ A_i do
    Aggregate(c_{i,j});
    CurCost ← CurCost + Cost(c_{i,j});
  end for
  if CurCost < Cost(c) then
    Cost(c) ← CurCost;
    Best(c) ← ⋃_{a_{i,j} ∈ A_i} Best(c_{i,j});
  end if
end for
```

5.4 Optimizing Space Cost with Bounded Query Processing Cost

The remaining question is how to choose a subset of cells to precompute, s.t.

(i) Any query can be answered successfully.

(ii) For any cell c, $Cost(c)$ is bounded by a user-specified threshold ϵ.

(iii) The storage cost (*i.e.*, the total number of precomputed cells) is as small as possible.

Since base cells can not be aggregated from other cells, they must be precomputed. For non-base cells, we define a topological order on these cells according to their granularity levels, *i.e.*, in the order, an r-D cell is put before an (r-1)-D cell. The intuition of how to select cells for precomputation is: we scan cells in the topological order one by one; for a scanned cell, we precompute cells as later as possible in the topological order. That is to say, we scan non-base cells one by one. For a cell c, if $Cost(c)$ does not exceed the threshold ϵ, we delay its computation to the online query processing, because the query time is still well bounded; otherwise, we materialize c. Such method is called $T - CUBING$, described in algorithm 2.

Algorithm 2 T-CUBING(c, ϵ)

```
ALGORITHM:
if c is a base cell then
  precompute c; Cost(c) ← 1;
else
  Aggregate(c) ;
  if Cost(c) > ε then
    precompute c; Cost(c) ← 1 ;
  end if
end if
```

The time complexity of T-CUBING for each cell is

$$O(N \max_i(|A_i|))$$

6. EXPERIMENTAL STUDY

ASRS (Aviation Safety Reporting System) [2] is a voluntary system run by NASA, that allows pilots and other airplane crew members to confidentially report aviation related safety incidents in the interest of improving air safety. An online system [3] [22] has built up to test the text cube ideas on the ASRS dataset. Several algorithms [13, 23, 7] are implemented in the system.

In our experiments, both a case study (Section 6.1) and a performance study (Section 6.2) are given. All algorithms are implemented in C++ (Visual Studio 2005) with SQL (Microsoft SQL Server 2008), conducted in a 0.99GHz CPU and 1G memory PC.

ASRS Dataset. 60,499 flight accident records that happened during the past ten years are downloaded from the ASRS database. Outliers are removed. Each record consists of a pilot report (*i.e.*, document) and 56 attributes, among

[2] http://asrs.arc.nasa.gov/
[3] http://inextcube.cs.uiuc.edu/nasa/

which we use 10 categorical attributes as the dimensions in our text cube. The 10 dimensions are 'Date', 'State', 'Person', 'Weather', 'Light', 'Engine Make Model', 'Flight Phase', 'Problem Primary Area', 'Event Anomaly Type' and 'Resolutory Action'.

Preprocessing. 39,272 words/phrases are extracted from pilot reports, and 100 topics are generated by Latent Dirichlet Allocation [2]. A text cube is built on the ASRS dataset, which contains 16.67 trillion cells, among which 1,677,587 are non-empty cells.

6.1 Case Study

Fatigue is defined as 'a non-pathologic state resulting in a decreased ability to maintain function or workload due to mental or physical stress.' Fatigue is a threat to aviation safety because of the impairments to alertness and performance it creates, which is a normal response to many conditions common to flight operations because of sleep loss, shift work, and long duty cycles [4, 14, 9].

6.1.1 Analysis of Relevant Topics

Given the keyword query ('fatigue', 'tired'), we calculate the relevance score for each topic by Equation 2. Topic 85 is the one with the highest relevance score, whose word distribution is shown in Table 4.

day	0.0888	hour	0.0619	trip	0.0417
time	0.0416	duty	0.0346	rest	0.0301
night	0.0288	minute	0.0279	leg	0.0258
fatigue	0.0182	late	0.0156	schedule	0.0150
morning	0.0140	long	0.0138	day	0.0133
fly	0.0124	early	0.0115	tired	0.0111
sleep	0.0104	previous	0.0102	hotel	0.0092
crew	0.0088	period	0.0088	arrive	0.0087
home	0.0079	legal	0.0072	block	0.0062
total	0.0041	evening	0.0041	delay	0.0041
work	0.0040	leave	0.0038	break	0.0038
assignment	0.0038	overnight	0.0037	reserve	0.0037
desk	0.0036	sick	0.0036	layover	0.0029
body	0.0029	month	0.0028	reduce	0.0027
show	0.0026	afternoon	0.0026	sequence	0.0024
company	0.0023	pair	0.0022	depart	0.0022
international	0.0022	begin	0.0021	room	0.0021
factor	0.0020	week	0.0020	pick	0.0019
assign	0.0018	deadhead	0.0018	wait	0.0018
bed	0.0018	awake	0.0018	flight	0.0017

Table 3: The Word Distribution of Topic 'Fatigue'.

Below is an interesting pilot report that talked about 'fatigue', which mentioned several words in Table 4 such as 'fatigue', 'sleep', 'hour', 'rest', 'depart', 'leg', 'break', 'duty', 'day', *etc.*:

EXAMPLE 4. *FATIGUING ASSIGNMENTS. AFTER I LNDG IN ZZZ I WENT TO SLEEP AT XA00 ZZZ1 TIME. MY PRE ALL-NIGHTER NAP WAS AT XN00 ZZZ1 TIME. MY POST ALL-NIGHTER REST WAS AT XD00 ZZZ1 TIME AND MY REST BEFORE AN XA00 LAX DEP WAS AT XD00 ZZZ1 TIME. THAT IS 4 DIFFERENT SLEEP TIMES IN LESS THAN 48 HRS. UPON LNDG IN ZZZ1 I WAS EXPECTED TO DO 2 MORE LEGS WITH 2 HR BREAKS FOR A 12+ HR DUTY DAY ON DAY 5! THIS WOULD HAVE BECOME UNSAFE AND I CALLED IN FATIGUED.*

6.1.2 Analysis of Correlated Topics

The topic 85 shown in Table 3 describes general terms related to 'fatigue', from which we can clearly infer that 'long duty' and 'insufficient rest' are two major factors that cause 'fatigue'. However, many other reasons that lead to or related with 'fatigue' are not so obvious as topic 85. By mining correlated topics, we are able to discover more detailed, complex and various reasons for 'fatigue'.

Concretely, we enumerate all cells whose aggregated dimensions are no more than 2; for each cell, we compute the correlation score between topic 'fatigue' and other topics in that cell; finally, we rank topics (associated with cells) according to their correlation score (see Table 4 [4]). One row should be understood as: the topic t (*i.e.*, the 3^{rd} column) is correlated with topic 'fatigue' in the cell c (*i.e.*, the 2^{nd} column) with the correlation score beng s (*i.e.*, the 1^{st} column). A sample pilot report is given (*i.e.*, the 4^{th} column) as the supporting evidence, as well as human interpretation (*i.e.*, the 5^{th} column).

Table 4 actually shows the effectiveness of our approach, for example, on row 2, which has a relatively high correlation score, the most correlated topic has higher probabilities for the words 'stress', *etc.*. In the representative report, the pilot first explained he/she forgot something, then later actually mentioned it may be because of fatigue, and there was not enough rest between the flights. And the fact that this correlation is high in cell '[Flight Phase]: cruise level' may indicate pilots are most influenced by fatigue during that phase. On row 3, the correlated topic contains 'lack', 'focus', and also 'fatigue'; and it is also common sense that in '[Weather]: Fog', the pilots or drivers are easily tired, and thus lost focus.

6.2 Performance Study

6.2.1 Experiment 1: Storage Size

Figure 1 reports the storage costs while varying (i) the threshold ϵ in the T-CUBING algorithm, and (ii) the number of dimensions of the text cube. FULL is the cube with full materialization, and CUBE20, CUBE60 and CUBE100 are the text cubes with ϵ being 20, 60 and 100, respectively. The number of dimensions varies from 2 to 10.

We can observe and/or verify two facts: (i) In principle, the smaller ϵ is, the more cells need to be pre-computed, thus leading to a larger storage cost. As verified in Figure 1, FULL always has the largest storage size, since FULL is equivalent to a text cube with ϵ being 1. To the opposite,

[4]For the first row, second column, * means the cell that aggregates all dimensions

Score	Discovered Cell	Correlated Topic	Sample Document	Human Interpretation
1.000	*	day, hour, trip, time, duty, rest, night, leg, fatigue, min, late	*FLIGHT HAD PREVIOUSLY BEEN DELAYED AND WE HAD MINIMUM REST PERIOD COMING UP, LESS THAN 9 HOURS.*	**Duty Cycle.**
0.6092	[Flight Phase]: cruise level; [Resolutory Action]: equipment problem dissipated	year, good, time, month, experience, fly, past, stress	*ALTHOUGH I AM COMPLETELY FAMILIAR WITH THE AIRSPACE; I COMPLETELY FORGOT ABOUT THAT SEGMENT OF THE CLASS B*	**Attitude**
0.5509	[Weather]: fog ; [Resolutory Action]: issue new clearance	awareness, failure, attention, realize, focus, fatigue, lack	*AS WE FLEW FURTHER OUT OVER THE WATER; THE CLOUDS SEEMED TO BE SLIGHTLY LOWER IN PLACES*	**Illusion**
0.5312	[Resolutory Action]: took evasive action; [Make Model]: airbus	pos, radar, supervise, trainee, CTLR, error, alert, busy, sector	*I THINK HE CONFUSED 10:00 POS AND 2:00 POS; AS THEY ARE BOTH 20 DEGREES OFF OF OUR NOSE.*	**Proficiency**
0.5085	[Flight Phras]: landing; [Event Anomaly]: landing without clearance	time, high, workload, unable, lack, delay, difficulty, additional	*I NEGLECTED TO RESELECT THE OTHER SIDE OF THE RADIO TO TALK TO TWR AS I WAS BUSY WITH THE CHKLIST.*	**Taskload**

Table 4: Correlated Topics with the Topic 'Fatigue'

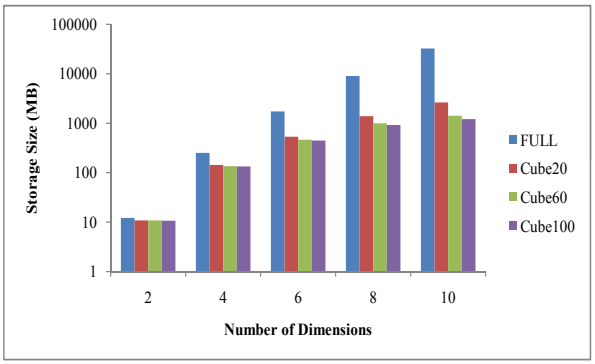

Figure 1: Storage Cost

CUBE100 is always the smallest one. (ii) The space compression ratio from BASIC to CUBE20 is larger than the one from CUBE20 to CUBE60, than the one from CUBE60 to CUBE100. Although the storage size is monotonically reduced when ϵ increases, such reduction becomes trivial when ϵ is sufficiently big. How to select an appropriate ϵ to balance the time and the space costs is still an interesting question left for future work. (iii) The compression ratio increases when the number of dimensions increases. The reason is that the text cube with more dimensions has a smaller average number of documents in cells, which results in less pre-computation.

6.2.2 *Experiment 2: Query Processing Time*

We report the query processing time in text cubes with different threshold (*i.e.*, CUBE 20, CUBE60 and CUBE100). BASIC is the baseline query processor which computes measures by retrieving documents in the raw database.

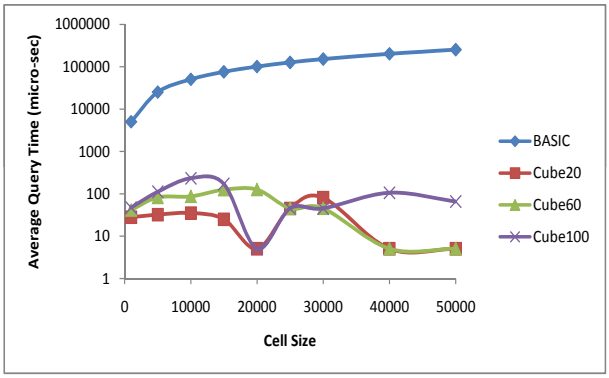

Figure 2: Query Time by Varing Cell Size

In Figure 2, the average query processing time is shown as a function of cell size (the size of a cell is the number of aggregate documents in the cell). As expected, BASIC increases its processing time approximately linearly, while other curves vibrate as cell size increases. The reason can be explained from the materializing procedure of T-CUBING: at the very beginning, all base cells are precomputed; as

the cell size increases, more and more cells need to be accessed to answer queries; when $Cost()$ reaches the threshold ϵ, T-CUBING begins to precompute cells again.

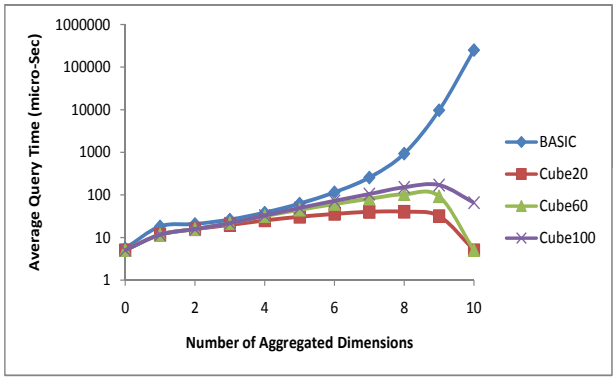

Figure 3: Query Time by Varing # Dimensions

In Figure 3, the average query processing time is plotted as a function of the number of aggregated dimensions of the queried cell. As expected, BASIC increases its query time sharply, while CUBE20, CUBE60 and CUBE100 rise comparatively smoothly with the help of the text cube, among which CUBE20 is the fastest. The similar vibration behavior happened as in Figure 2.

7. CONCLUSIONS

We demonstrated a novel method to generate correlated topic models from a large corpora of text reports which can be analyzed in the subspaces defined by cells defined by the intersection of numerous fixed fields and discovered correlated topics. The research is motivated by the need to develop technologies to help uncover aviation safety incidents before they happen based on large repositories of aviation safety narratives. These narratives are also annotated with numerous fixed fields, thus giving an excellent application domain for this research. The large number of potential cells in the resulting text cube demand that the computational complexity be sufficiently bounded. We applied this novel system to the analysis of crew fatigue and show potential factors that may be related to fatigue issues. Although a full analysis of crew fatigue and its contributing and correlated factors is out of the scope of this paper, the technologies described can be used in future studies to understand these issues.

8. ACKNOWLEDGMENT

The work was supported in part by HP Labs, NASA grant NNX08AC35A, the U.S. National Science Foundation grants IIS-09-05215, and the NASA Aviation Safety Program.

9. REFERENCES

[1] D. M. Blei and J. D. Lafferty. Correlated topic models. In *NIPS*, 2005.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. In *NIPS*, pages 601–608, 2001.

[3] A. Broderick, P. Bevilaqua, and et. al. Wake turbulence: An obstacle to increased air traffic capacity. Technical report, National Research Council, 2008.

[4] J. Caldwell. Fatigue in the aviation environment: An overview of the causes and effects as well as recommended countermeasures. In *Aviation Space and Environment Media*, 1997.

[5] L. Connell. Aviation safety reporting system. Technical report, NASA Ames Research Center, 2010.

[6] B. Ding, D. Lo, J. Han, and S.-C. Khoo. Efficient mining of closed repetitive gapped subsequences from a sequence database. In *ICDE*, pages 1024–1035, 2009.

[7] B. Ding, B. Zhao, C. X. Lin, J. Han, and C. Zhai. Topcells: Keyword-based search of top-k aggregated documents in text cube. *ICDE*, 2010.

[8] J. Gray, A. Bosworth, A. Layman, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-total. In *ICDE*, pages 152–159, 1996.

[9] B. M. Institute. A review of issues concerning duty period limitations, flight time limitations, and rest requirements. *Federal Aviation Administration, Washington, D.C*, 1998.

[10] J. P. Jiawei Han, Micheline Kamber. Data mining: Concepts and techniques. In *Morgan Kaufmann*, 2005.

[11] D. Jones and M. Endsley. Sources of situation awareness errors in aviation. In *Aviation, Space, and Environmental Medicine*, pages 507–512, 1999.

[12] V. Lavrenko and W. B. Croft. Relevance-based language models. In *SIGIR*, pages 120–127, 2001.

[13] C. X. Lin, B. Ding, J. Han, F. Zhu, and B. Zhao. Text cube: Computing ir measures for multidimensional text database analysis. In *ICDM*, pages 905–910, 2008.

[14] NTSB. Safety study: A review of flight crew involved, major accidents of u.s. air carriers, 1978-1990. *Washington, D.C: NTSB*, 1994.

[15] N. Oza, J. P. Castle, and J. Stutz. Classification of aeronautics system health and safety documents. *IEEE Transactions on Systems Man and Cybernetics Part C 39:670-680*, 2009.

[16] K. Salomatin, Y. Yang, and A. Lad. Multi-field correlated topic modeling. In *SDM*, pages 628–637, 2009.

[17] F. Song and W. B. Croft. A general language model for information retrieval (poster abstract). In *SIGIR*, pages 279–280, 1999.

[18] A. N. Srivastava and B. Zane-Ulman. Discovering recurring anomalies in text reports regarding complex space systems. *2005 IEEE Aerospace Conference Proceedings*, 2005.

[19] M. Turchi, A. Mammone, and N. Cristianini. *Text Mining: Classification, Clustering, and Applications*, chapter Analysis of Text Patterns using Kernel Methods, pages 1–22. CRC Press, 2009.

[20] Unknown. Acn 704835. Technical report, Aviation Safety and Reporting System, 2006.

[21] X. Wang, C. Zhai, X. Hu, and R. Sproat. Mining correlated bursty topic patterns from coordinated text streams. In *KDD*, pages 784–793, 2007.

[22] Y. Yu, C. X. Lin, Y. Sun, C. Chen, J. Han, B. Liao, T. Wu, C. Zhai, D. Zhang, and B. Zhao. inextcube: Information network-enhanced text cube. *PVLDB*, 2(2):1622–1625, 2009.

[23] D. Zhang, C. Zhai, and J. Han. Topic cube: Topic modeling for olap on multidimensional text databases. In *SDM*, pages 1123–1134, 2009.